

## GA による動的マッピングの PDES への適用

三橋 純, 永田 和敏, 成田 誠之助  
早稲田大学理工学研究科

PDES の実行効率に影響を与える要素として、仮想時刻同期手法とマッピング手法が挙げられる。

本論文では、動的マッピングの手法として遺伝的アルゴリズム (GA) を用いた動的マッピングを提案する。シミュレーションの進行と平行して、ホストプロセッサではセルプロセッサから送られてきたデータをもとに GA の計算を行う。その結果を定期的にセルプロセッサへと送信し、動的にマップを更新していく。

また、仮想時刻同期手法として、同期的手法において他の PE に隣接しているノードと隣接していないノードを区別し、保証時刻を長くとることにより同期回数を軽減する。

これら二つのアプローチによりシミュレーションの高速化を目指す。

## Application of dynamic mapping with Genetic Algorithm to Parallel Discrete Event Simulation

Jun Mitsuhashi, Kazutoshi Nagata, Seinosuke Narita  
Department of science and engineering, Waseda University

There are two major elements which influence the execution efficiency of PDES. One is a synchronization method, and the other is a mapping method.

This thesis proposes a dynamic mapping using Genetic Algorithm (GA). The host processor calculates maps using the data from cell processors with GA algorithm in parallel with the progress of the simulation. The result is regularly transmitted to cell processors, and the map is updated dynamically. This dynamic processing enables high-speed processing of the simulation.

### 研究背景

年々、ネットワークシステムを初めとする様々なシステムが大規模化、複雑化の道を進んできている。それにより、シミュレーションの分野においても大規模なモデル、複雑なモデルを扱うものが増え、その計算時間の短縮というのが大きなテーマの一つとして挙げられるようになった。シミュレーションを高速に行うための方法として、コンピュータの処理速度を向上させることが考えら

れるが、これは費用および時間に多大なコストを要し、また、その時代における限界というのがある。そこで、複数のコンピュータを用いて並列処理を行う並列シミュレーションが注目を浴びるようになった。その中でも特に離散事象シミュレーションを扱ったものを並列離散事象シミュレーション (PDES) といい、我々、成田研究室シミュレーション班は、この PDES の高速化を目的として研究を行ってきた。

## PDES

並列離散事象シミュレーションにおいて、その実行効率に影響を与える要素として仮想時刻同期手法およびマッピング手法が挙げられる。仮想時刻同期手法はシミュレーションの正当性を保ちながらデッドロックに陥ることなく効果的にメッセージを送るための手法である。また、マッピング手法は各ノードをプロセッサに割り当てるための手法であり、事象処理数の均等化、および、メッセージ数の削減を目的とする。

仮想時刻同期手法として、従来の同期的手法において、保証時刻を決定する際、他の PE へと隣接するノード、すなわち PE 間のメッセージを発生する可能性のあるノードと、他の PE へと隣接しないノードを区別することで、保証時刻を長くとり、それにより同期回数の軽減を図り、シミュレーションの高速化を目指す。

また、マッピング手法においては遺伝的アルゴリズムを用いた動的マッピング手法を適応した。セルプロセッサはシミュレーションの進行に伴い、定期的にデータをプロセッサへと送信する。ホストプロセッサではそのデータを用いて新たなマップを遺伝的アルゴリズムにより作成し、やはり定期的にセルプロセッサへ送信することで動的にマップを更新していく。

## 環境

本研究における環境として、

ホストプロセッサ

MMX Pentium233MHz (メモリ 128MB) セルプロセッサ (4台)

Pentium II 450MHz (メモリ 384MB)

によって構成された PC クラスタを用いた。なお、それぞれの PC はスイッチングハブを介して 100Base-TX によって接続されており、通信ライブラリとしては、現在、多くのプラットフォーム上で使用可能である MPI (Message Passing Interface) を用いた。対象モデルとしては 1024 ノードよりなる待ち行列システムシミュレータを用いた。

## GA による動的マッピング

遺伝的アルゴリズム (GA) は、生物の進化プロセスをモデルにした、進化的計算法の一つであり、解きたい問題を環境に、問題の解を環境に存在する個体に対応させ、それらを進化させることにより優れた解を求める手法である。

遺伝的アルゴリズムは以下のような手順からなる。

- 1: 初期集団の生成
- 2: 適応度の計算
- 3: 終了判定
- 4: 個体の選択・淘汰
- 5: 交叉
- 6: 突然変異
- 7: 世代交代 (2へ戻る)

これを PDES へと適用すると以下ようになる。

まず、初期マップをファイルから読みこみ、それを個体数分コピーする。全ての個体について適応度を計算し、そのうち、適応度の高い 10% を選択し、他は淘汰する。適応度の算出法については後に示す。親個体として選ばれたものから任意に 2 個体を選び、ある点で交叉させ、二つの新たな個体を作り出す。これを設定した個体数になるまで繰り返す。得られた個体に対し突然変異をおこさせる。突然変異に関しても後に説明する。これまでの作業により得られた個体を新たな親世代として世代交代をし、適応度の計算に戻って繰り返し計算は続いていく。

適応度はシミュレーション中の各セルプロセッサから統計データを定期的に集計し、そのデータによって計算される。シミュレーションの効率を向上させるためには、PE 間のメッセージ数の軽減および各 PE における事象処理数の均等化が指標となる。セルプロセッサはシミュレーションの進行に伴い、定められた同期回数毎に各ノードにおける事象処理数、および各リンクを通ったメッ

セージ数をホストプロセッサへと送出する。

$$\text{通信コスト} = (\text{PE 間のメッセージ数}) / (\text{総メッセージ数}) \times 1000$$

$$\text{事象コスト} = ((\text{PE の事象処理数}) - (\text{平均事象処理数}) / (\text{平均事象処理数}))^2 \times 1000$$

$$\text{適応度} = \text{通信コスト} + \beta \times \text{事象処理コスト}$$

(β: パラメータ)

によって求められた適応度により、選択・淘汰が行われる。この計算法により、通信コストと事象コストが同じオーダーになる。さらに、シミュレーションをする環境の変化にはβを設定することで対応する。

次に終了判定であるが、シミュレーションの終了に伴い、ホスト側の計算も終了するので適応度による終了判定は用いない。

今回、突然変異は2通りの突然変異を用意した。まず一つは乱数によりあるノードを選びそのノードに隣接するノードおよび、さらにその隣接ノードへと隣接するノードを選ばれたノードと同じ PE へと変更する。もう一つは、個体の領域の数を算出し、使用 PE 台数よりも大きい時には、最も小さな領域をその周りの領域と結合させるというものである。前者の方法により、ランダムなマップからスタートしたとしても、領域数は比較的高速に減少していくと思われる。また、後者の方法は、領域数が使用 PE 台数よりも多いという局所解へと陥るのを避けるためである。

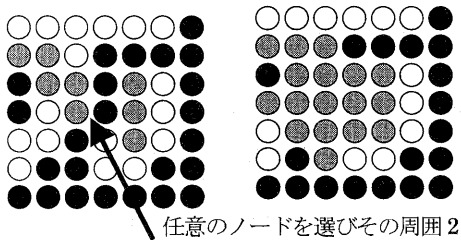


図1 突然変異1

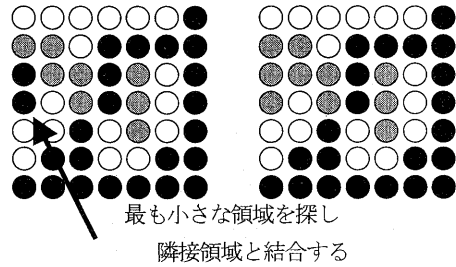


図2 突然変異2

以上のアルゴリズムをシミュレータへと適用した。なお、モデル1, 2は2次元トラスメッシュ状のモデルであり、初期マップとしてランダムマップを用いたものがモデル1、位置情報により均等に4分割されたマップを用いたものがモデル2である。モデル3, 4は、ランダムに1024個のノードを配置し、近隣のノードにリンクをはったものであり、初期マップとしてランダムマップを用いたものがモデル3、位置情報により均等に4分割されたマップを用いたものがモデル4である。それぞれのモデルについて、実行時間、総メッセージ数、最大事象処理PEと最小事象処理PEの処理数の差をグラフにした。

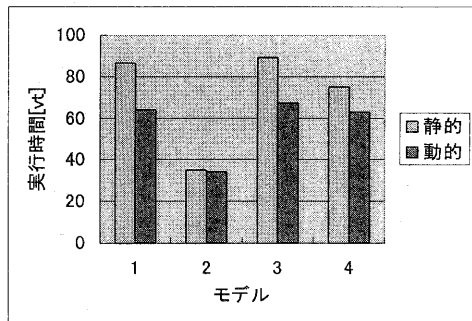


図3 動的マッピングによる性能

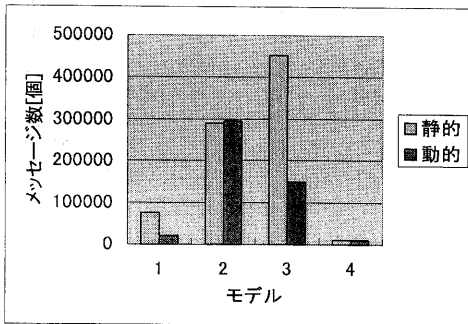


図4 メッセージ数の比較

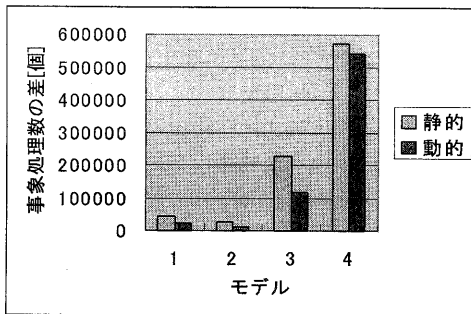


図5 事象処理数の差

得られた結果を見ると、実行時間、メッセージ数、事象処理数の格差、全てにおいて動的な処理による向上が見られる。特に、モデル 2, 4 については、最適に近いと思われる位置情報によるマップを用いているが、それに対しても性能が向上している。PDES のマッピング問題はメッセージ数、事象処理数の両方を考慮しなければならないためにマップを算出するのは難しい。しかし、この GA を用いたアルゴリズムはいくつかのパラメータを設定するだけであとはシミュレーションの進行中により適したマップへと進化していくので、動的マッピングの一つの手段として有効であるといえよう。

#### 仮想時刻同期処理

PDES では、事象を生起した場所により各 PE に割り当ててシミュレーションが進められていく。各 PE は事象処理中に他の PE で処理されるべき事象が発生した場合、それをメッセージとして送

信しなければならないが、そのタイミングを誤ると事象処理の逆転が起これシミュレーションの信頼性をなくすことになる。現在まで、これらの制御手法として保守的手法では、Null Message 法、同期法が提案され、楽観的手法では、Time Warp 法などが提案されてきた。本研究では、これまでの研究において比較的高速な処理が実現可能と位置づけられている同期手法をベースに検討し、その改良型である改定同期法を仮想時刻同期手法に適用した。

#### 同期法

同期法とは、仮想時刻同期の際にバリア同期を行い、他のすべての PE に対する保証時刻を同時に計算する。各 PE は、計算されたその保証時刻までの事象を他の PE に関係なく処理できる。この保証時刻算出方法は次の通りである。

- ・まず現在の仮想時刻を  $T_c$  (先頭事象時刻) とすると、すべての PE において各 PE が独立して事象処理が行える時刻  $T_s$  を求める。

$$T_s = T_c + T \quad (T: \text{最小保証時間})$$

- ・次に各 PE で  $T_s$  の最小値  $T_{s_i}$  を求める。

$$T_{s_i} = \min(T_{s_1}, \dots, T_{s_q})$$

この時刻までは、各 PE は独立して事象処理を行うことができる。以降この作業の繰り返しで、保証時刻は算出されていく。

#### 改定同期法

これまでの我々の研究から並列シミュレーションにおいて PE のバリア同期コストがメッセージ送信コストに比べても無視できないほど大きいことがわかっている。よって今回この改定同期法では、同期法の保証時刻算出を改定し同期間隔を延長、そして、同期コストを減少させることがねらいである。また、同期回数を減らすことにより他の PE に送信するメッセージも一括送信が可能となりメッセージ送信コスト削減も実現できる。

さて、前出の通り同期法における次の同期までの保証時刻算出はすべての PE において先頭事象時刻に最小保証時間をあわせたものの最小値であ

った。しかしこの最小保証時間というものは、PE間でのトランザクションの移動において、考えられる所用時間の最小値である。つまり最小保証時間は、メッセージが起こった場合に初めてその役目を果たすものといってよい。そこで、改定同期法における保証時刻算出方途は、事象リストからメッセージに関する事象を探しだしその最早起メッセージ事象に最小保証時間を足すことによって従来よりも長く保証時間を定義することが可能である。しかしこれによって、同期コストが削減できたとしても、新アルゴリズムによって派生してくる事象リストからのメッセージ事象抽出に関するコストは、馬鹿にはできない。そこで、保証時刻算出としては多少弱くなるが、事象リスト走査コストを軽減するアルゴリズムの方が好ましい

#### アルゴリズム

図6のようにマッピングされたデータを考える。メッセージに関する事象とは、各PEにマッピングされた境界に存在するノードにおける事象である。今図7のような事象リストを見ると、まず最初にノードAでの事象が処理される。このノードは境界ノードから二つ分離れているためメッセージ事象へと変化するのは、最低2回以上の事象処理が必要であるが、先読みするコストのことを考えて境界を除くすべてのノードの事象に対して、事象の予測は1回のみと定義した。つまり、事象Aで処理されたトランザクションは次の事象では、境界ノードに存在することになる。しかしこの走査は、保証時刻算出のために行っていることであり何らシミュレーションの信用性を損なうものではない。この保証時刻算出の様子を図7に示す。

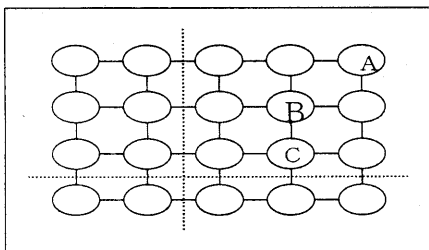


図6 メッセージ発生ノード

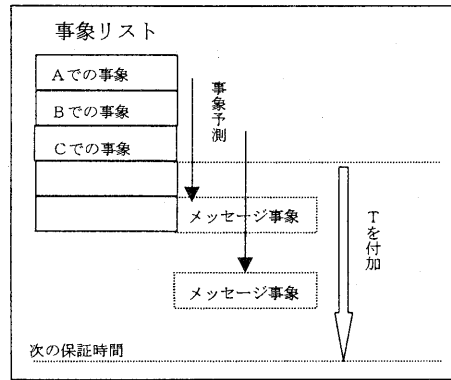


図7 改定同期アルゴリズム

以上の手法を前出のモデル1~4に適用した結果を図8, 9に示す。

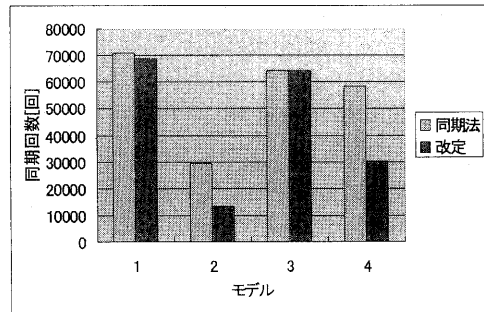


図8 同期回数の推移

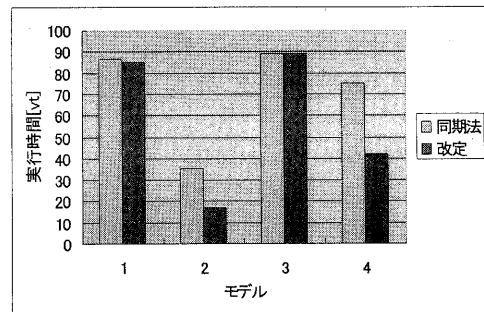


図9 改定同期法の実行時間

得られた結果を見ると、最適なマップを用いたモデル(モデル2, 4)において、同期回数の減少と比例して実行時間も減少していく様子が見られる。しかしランダムにノードを割り振ったモデル

(モデル 1、3)においては、同期回数はやや減少しているものの、実行時間が長くなるモデルも見られた。これは、ランダムマッピングを用いた結果、その事象リスト中のメッセージ事象が莫大に増加してしまい効果が見られなくなったものと思われる。この手法はメッセージ事象に影響が左右されやすく、あらかじめ性能の上がるマッピングを用意しておくのが好ましいと思われる。

#### 手法の統合

これら二つの手法を用いて、統合的なシミュレータを作成した。その結果を以下に示す。

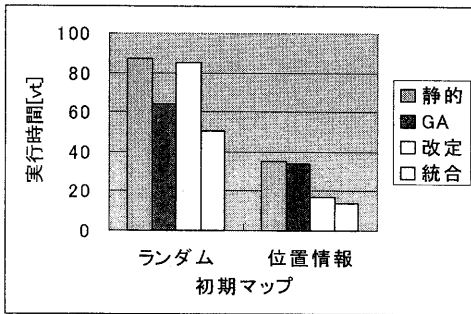


図 10 2次元トーラスメッシュの実行結果

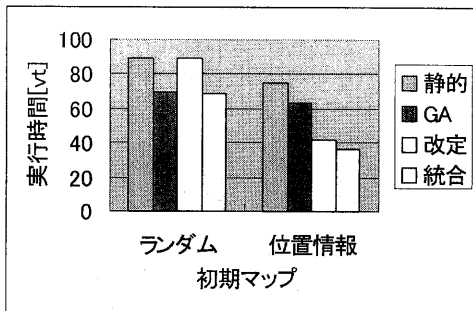


図 11 ランダムモデルの実行結果

図 10, 11 からわかるように全ての場において、統合的なシミュレータが最も効率の良いシミュレーションを行っている。前述した通り、ランダムなマップにおいては、改定同期法の効果は見られないが、GA による動的マッピングと統合することでシミュレーションの進行に伴い、その効果が次第に現れてくるのがわかる。比率でいうと、

それぞれ 173%, 253%, 131%, 206%への性能の向上が見られた。また、メッセージ数、事象処理数の格差、同期回数についてはそれぞれ単独に適用した時と同様に、GA による動的マッピングにより、メッセージ数、事象処理数の格差が減少されており、改定同期手法により、同期回数の削減が実現できている。以上のことから、今回実装したこれら二つの手法は有効な手法であるといえよう。

#### 結論

今回、仮想時刻同期手法およびマッピング手法の二つのアプローチから PDES の高速化を目指した。静的なマップを用い、同期的手法を用いたものに比べると、大きな性能の向上が見られた。また、マッピング手法においては、最適に近いと思われる位置情報による分割と比べても、性能の向上が得られたので効果的な手法であると考えられるであろう。また、改定同期法においては、ある程度効率の良いマップと共に用いないと大きな効果が得られないが、GA による動的マッピングと併用することでその欠点も解消できる。以上のことから、今回提案するこれら二つの手法による統合的なシミュレータは効果的なものであると考えられる。

#### 参考文献

- [1] D. M. Nicol. Principles of conservative parallel simulation. In 1996 Winter Simulation Conference, PP. 128-135, 1996
- [2] Samir R. Das. Adaptive protocols for parallel discrete event simulation. In 1996 Winter Simulation Conference, PP186-192, 1996
- [3] R. M. Fujimoto. Parallel and distributed discrete event simulation: Algorithms and applications. In 1993 Winter Simulation Conference, pp. 106-114, 1993
- [4] J. Holland. "Adaptation in Natural and Artificial Systems"
- [5] 高井 峰生. 離散事象システムの効率的並列シミュレーションに関する研究. 博士論文, 1997